

Programma nima?

Programma bu aniq bir muammoni yechish uchun ko'rsatmalar ketma keltigidir. Programmalash esa mana shu ko'rsatmalarni topish. Shuning uchun, programmist uchun eng muhim kerak bo'ladigan mahorat bu berilgan muammoni hal qilish ketma-ketligini o'ylab topish. Yechimning aniq va to'g'ri bo'lishligi shart. Bu mahorat, programma tuzish orqali shakllantiriladi. Qancha ko'p programma tuzsangiz shuncha yaxshi, ko'p o'qish yoki kino ko'rish bilan emas :).

Programma oddiydan juda murakkab bo'lishi mumkin: masalan sonlarni qo'shish yoki xujjatdan kerakli so'zlarni qidirish yoki rasm va videolarga ishlov berish va hokazo. Programma quyidagi oddiy ko'rsatmalardan tashkil topishi mumkin:

- Kiritish: klaviaturadan.
- Chiqarish: Malumotlarni ekranga chiqarish.
- Hisoblash: Amallarni bajarish, qo'shish, ayirishga o'xshash.
- Tekshirish: Kodni to'g'riligni tekshirish.
- Takrorlash: Biron-bir ko'rsatmani qayta-qayta qilish.

Quvonarlisi shundaki, deyarli shu ko'rsatmalar siz ishlatgan yoki siz qilmoqchi!

bo'lgan programmalarda bo'ladi. O'shaning uchun, programmalashni biz katta topshiriqni kichikdan kichik topshiriqlarga shunday bo'lamizki -- kichik topshiriqlar yuqorida ko'rsatilgan oddiy ko'rsatmalar bilan hal qilinadigan bo'lsin.

Pythonni ishlatish

Hozirda har bir operatsion sistemada python **interpretator** mavjud. Kompyuteringizdagi terminalni ishga tushiring va python deb yozingda `Enter` ni bosing. Va quyidagini ko'rishingiz mumkin.

```
Python 3.8.5 (default, Jan 27 2021, 15:41:15)
[GCC 9.3.0] on linux
Type "help", "copyright", "credits" or "license" for more information.
>>>
```

`>>>` belgisi 'prompt' deb ataladi va u python interpretatori tayyorligini bildiradi. Boshqa yozuvlar hozircha muhim emas. Endi programma tuzishga marhamat!

Birinchi programma

Promptga quyidagini yozing

```
>>> print("Salom, Pythonjon!")
```

Keyin `Enter` ni bozsak, quyidagini ko'rishingiz mumkin,

```
>>> print("Salom, Pythonjon!")
Salom, Pythonjon!
```

Dehqon tilida, kompyuterga `Salom, Pythonjon!` yozuvini chiqargin deb ko'rsatma (kommanda) berayapmiz. Bu yerda biz `print` dan foydalanayapmiz. `print` python tiliga tegishli so'z bo'lib ekranga yozuvlarni chiqarishda ishlatiladi.

Umuman olganda,

Dehqonchasiga	Pythonchasiga
Kompyuterga "Salom, Pythonjon!" yozuvini chiqargin	<code>print("Salom, Pythonjon!")</code>

Oddiy arifmetika

Prompt kuchli narsa. Misol uchun biz u yerda oddiy arifmetik amallardan tortib murakkab amallarni bajarishimiz mumkin. Masalan kundalik hayotimizda sonlar bilan ko'p ishlatamiz. Ularni qo'shamiz, ayiramiz, bo'lamiz ko'paytiramiz va hokazo. Xo'sh bu amallarni kompyuterda qanday qilamiz? Aniqrog'i pythonda qanday qilamiz? Umumiy qilib mazkur amallar **operator**lar deb ataladi. Arifmetik amallar uchun quyidagi operatorlar mavjud:

Dehqonchasiga	Pythonchasiga	Inglizcha
qo'shish	+	Add
ayirish	-	Subtract
ko'paytirish	*	Multiply
bo'lish	/	Divide
darajaga oshirish	**	Exponent
butunli bo'lish	//	Floor divison
qoldikli bulish	%	Modulus

Misollar,

```
>>> 2+2
4
>>> 6-2
4
>>> 2*2
4
>>> 3**2 + 6
15
>>> 4/5
0.8
>>> 4//5
0
```

Qiymat va turlar

Qiymat bu programma bilan ishlaydigan muhim bo'g'in. Hozirgacha ko'rgan barcha "Salom, Pythonjon", 4, va 0.8 lar qiymatga misollardir. Ular bir-biridan turiga (type inglizchada) nisbatan farqlanadi. Xususan,

Qiymat	Turi (type)
"Salom, Pythonjon"	belgilar to'plami (string)
4	butun son (integer)
0.8	haqiqiy son (float)

"Salom, Pythonjon" bu `String` ga mansub. String bu ketma-ket kelgan harf yoki belgilardan tashkil topgan to'plamdir. `4` bu butun son turiga mansub (integer inglizchada). 0.8 bu haqiqiy son (float). Agar siz biron qiymatning qaysi turga tegishligini bilmoqchi bo'lsangiz quyidagicha qilasiz. Masalan, kompyuterda `4` qaysi turga tegishliligni aniqlash uchun pythonga tegishli `type` so'zidan foydalanamiz:

```
>>> type(4)
<class 'int'>
>>> type(0.8)
<class 'float'>
>>> type("Hello, World")
<class 'str'>
```

Natijalarga qarasak, 4 ga int ko'rsatilayapti. Bu butun son (integer inglizchada). Bu yerda `class` turi degani. 0.8 floatni ko'rsatayapti yani haqiqiy son. `'Salom, Pythonjon'` str (yani string qisqartirilgani) turiga tegishli.

E'tibor beringki biz qiymatlarni quyidagicha yozishimiz ham mumkin '4' yoki '0.8'. Bular songa o'xshaydi lekin ular qo'shtirnoq ichiga yozilgan o'shaning uchun ular string turiga mansub.

```
>>> type('4')
<class 'str'>
>>> type('0.8')
<class 'str'>
```

Turlar haqidagi bilimlarimizni quyidagi jadvalda jamladik:

Tur (Type)	Inglizcha	Namuna
Butun son	int	-5, -1, 0, 10, 999
Haqiqiy son	Float	2.3, -5.1, 100.25
String	String	'salom', 'okay', 'ha'

Qolaversa quyidagicha kod yozishimiz noto'g'ri

```
>>> "Hello, World!" + 1
```

Hamishi amallar bajarilayotganda ularning turlari bir-biriga munosib bo'lishligi kerak.

Ona va python tili

Ona tilimiz o'zbek hisoblanadi, yoki inglizlarning ona tilisi ingliz tilidir. Bu tillar insonlar tomonidan isloh qilinmagan, balki vaqt o'tishi bilan shakllanib kelgan.

Rasmiy tillar esa insonlar tomonidan maxsus maqsadlar uchun o'ylab topilgan. Masalan, yo'l harakati uchun til mavjud, yani yo'lda ketayotganda biz har xil belgilarni ko'ramiz. Ularni o'zbekistonning qaysi joyiga bormang -- bir xil. O'zbek, ingliz yoki tojik millati bo'lsin hamma haydovchi uchun tushunarli. Bu yo'l tili insonlar tomonidan malum meyorlar asosida ishlab chiqilgan. Va yo'ldagi holatlarni haydovchiga ko'rsatma berib ('aytib') turadi.

Xuddi shunga o'xshash python ham rasmiy til bo'lib u insonlar tomonidan ishlab chiqilgan. Python tili -- inson va kompyuter o'rtasidagi aloqa uchun hizmat qiladi.

Masalan siz pythonga odamga gapirgandek gapirsangiz u hech narsa qilmaydi!

Rasmiy tillar qattiq qoidalarga ega. Masalan tezlikni *10* dan oshirmang degan\ joyga *100* deb yozib qo'ysak, nima bo'lishligini bilish qiyin emas! -- avariya. Ona tilida gapirilganda qat'iy qoidalar yo'q va xato qilinsa tezda to'g'irlab ketaverishimiz mumkin. Yoki kerakli malumotni qaytadan takrorlash qo'shimcha qo'shish muvin. Python tilida har bir qism aniq bo'lishi kerak. Kerakli so'zlarni to'g'ri yozish nuqta va vergullarni to'g'ri yozish talab qilinadi aks holda programma ishlamaydi -- xuddi sizni kompyuter tushunmadi.

Diqqat!

Umuman olganda, programmistlar bor vaqti berilgan muammoni kompyuterga tushuntirish bilan o'tadi, kompyuter shunchalik 'tentakki' unga juda aniq ketma-ketligda tushuntirish kerak.

Debugging

Programmistlar xato qilishadi. Xatolarni topib va ularni yechishni debugging (dibagging deb o'qiladi) deb ataymiz.

Masalan quyidagi programma ishlatsangiz:

```
>>> print2)
```

Ishlamaydi, va quyidagi xato chiqariladi

```
File "<stdin>", line 1
    print2)
      ^
SyntaxError: unmatched ') '
>>>
```

SyntaxError deb ko'rsatilayapti. Yani, biz etibor bersak, bitta qavsni tashlab ketganligimizni bilishimiz mumkin. Va, aslida programma quyidagicha bo'lishligi kerak edi.

```
>>> print(2)
```

Debugging qilish uchun ham malaka kerak, vaqt o'tishi bilan programma qilish davomida o'rganib ketaveramiz.

Va nihoyat python interpretatoridan `exit()` ni promptga yozish bilan chiqib ketamiz.

```
>>> exit()
```

PyCharmda ishlash

Shu paytga qadar biz `>>>` promptda ishladik. Biz hozir qilgan ishlarimizni. Pycharmda fayl yasab keyin unga quyidagicha yozamiz va uni ishlatamiz.

```
print("Salom ,Pythonjon!")
```

E'tibor bering bu yerda `>>>` belgisi yo'q.

Biz ikkala usuldan keng foydalanamiz. O'shaning uchun ikkalasini ham bilishim shart. O'rni kelganda ko'rasiz.

Foydali terminlar

- **Muammo yechish:** Bu maummoni ifodalab unga aniq yechim topishdir. Bu ingliz tilida problem solving deb ataladi.
- **Programma:** Berilgan aniq hisobni amalga oshirish uchun kerak buladigan ko'rsatmalar ketma-ketligidir.
- **integer:** butun sonlarni bildiradigan tur: 12, 0, -50.
- **float:** ratsional/irratsinal sonlarni bildiruvchi tur: 0.5, 3.143454.
- **string:** belgilar ketma ketligini ifodalash uchun tur: "Salom, Pythonjon".
- **Syntax:** programmada ifodalar yozilishining umumiy qoidalari.
- **bug:** programmadagi hato (bug inglizchada) deb ataladi.
- **debugging:** xatolarni topish va ularni to'g'irlash.
- **print:** python tiliga hos so'z. Qiymatni ekranga chiqarish uchun ishlatiladi.
- **Interpretator:** bu programma boshqa programmani olib uni ishlatadi.
- **prompt:** interpretator `>>>` belgisini chiqaradi va uning tayyorligini bildiradi.

Problem solving

Promptda (>>>) bajaring. Yuqoridagi programmalarni o'zingiz yana qilib ko'ring, bu safar o'zgarishlar bilan.

1. print ga oid. Qavslardan birini oling va programmani ishlating (Enterni bosasiz). Hamma qavsni olib tashlasakchi?

```
>>> print("salom")
```

2. print ga oid. Qo'shtirnoqlardan birini oling va programmani ishlating.

```
>>> print("salom")
```

3. Mana shunday qilib yozsak programmaga nima bo'ladi 4+-4? 4++4 chi?
4. python primpitida 1 deb yozing va Enterni bosing. Keyin, 01? 0099? Nima bo'ldi?
5. 2+2 ni ishlating. 2 2 ni ishlating. Nima bo'ldi?
6. "Salom, Pythonjon!" + 1 ni ishlating? ishlamasa nimaga ishlamaganligini ayting.
7. 6 soat 30 sekundni sekundga aylantiring.
8. 2 ni 10 darajasini 555 ga bo'ling.
9. Ekranga "Salom, Dunyo" deb chiqaring.
10. Ekranga "Salom, Dunyo ingliz tilida 'Hello World' degani" deb chiqaring

Pycharmni ishlatingda quyidagilarni bajaring:

1. Ekranga "Salom, Dunyo" deb chiqaring.
2. Ekranga "Salom, Dunyo ingliz tilida 'Hello World' degani" deb chiqaring.